



SORSE

International Series of Online Research Software Events

<https://sorse.github.io>

EESSI

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

Wed Nov 25th 2020

Kenneth Hoste (HPC-UGent, Belgium)

kenneth.hoste@ugent.be - @kehoste (Twitter) - @boegel (GitHub)



whoami

kenneth.hoste@ugent.be



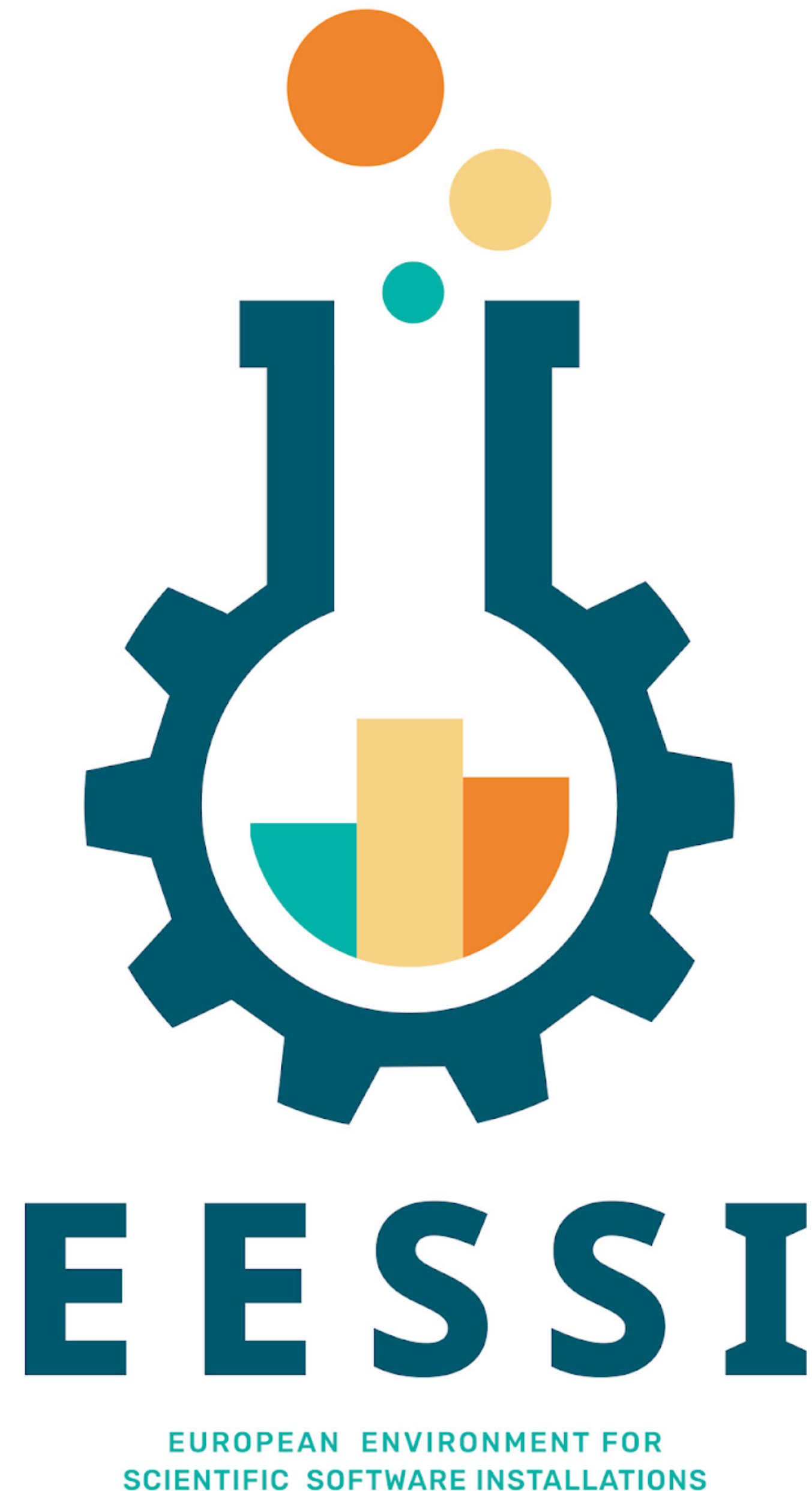
@kehoste



@boegel



- Computer scientist from Belgium
- Masters & PhD from Ghent University
- **HPC system administrator at UGent since 2010**
- User support & training, **software installations** (RSE?)
- Big fan of FOSS, family, loud music, (good) beer, dad jokes, stickers
- Doesn't like C++, CMake, SCons, Bazel, TensorFlow, OpenFOAM, ...



- **Who** is involved in EESSI?
- **What** is the project about?
- **Why** did we start it?
- **How** are we tackling the problem?
- What are the **challenges**?
- Which **FOSS projects** do we use?
- What is the **current status**?
- **Live demo!**

EESSI in a nutshell



- **European Environment for Scientific Software Installations**
(EESSI, pronounced as "easy")
- Collaboration between different partners in HPC community
- Goal: **building a common scientific software stack**
for HPC systems & beyond (cloud, workstations, ...)
- Grass roots project

<https://eessi-hpc.org>

<https://github.com/eessi>

<https://eessi.github.io/docs/pilot>

 *[@eessi_hpc](#)*

Project partners



UNIVERSITY OF TWENTE.



UiO : **University of Oslo**



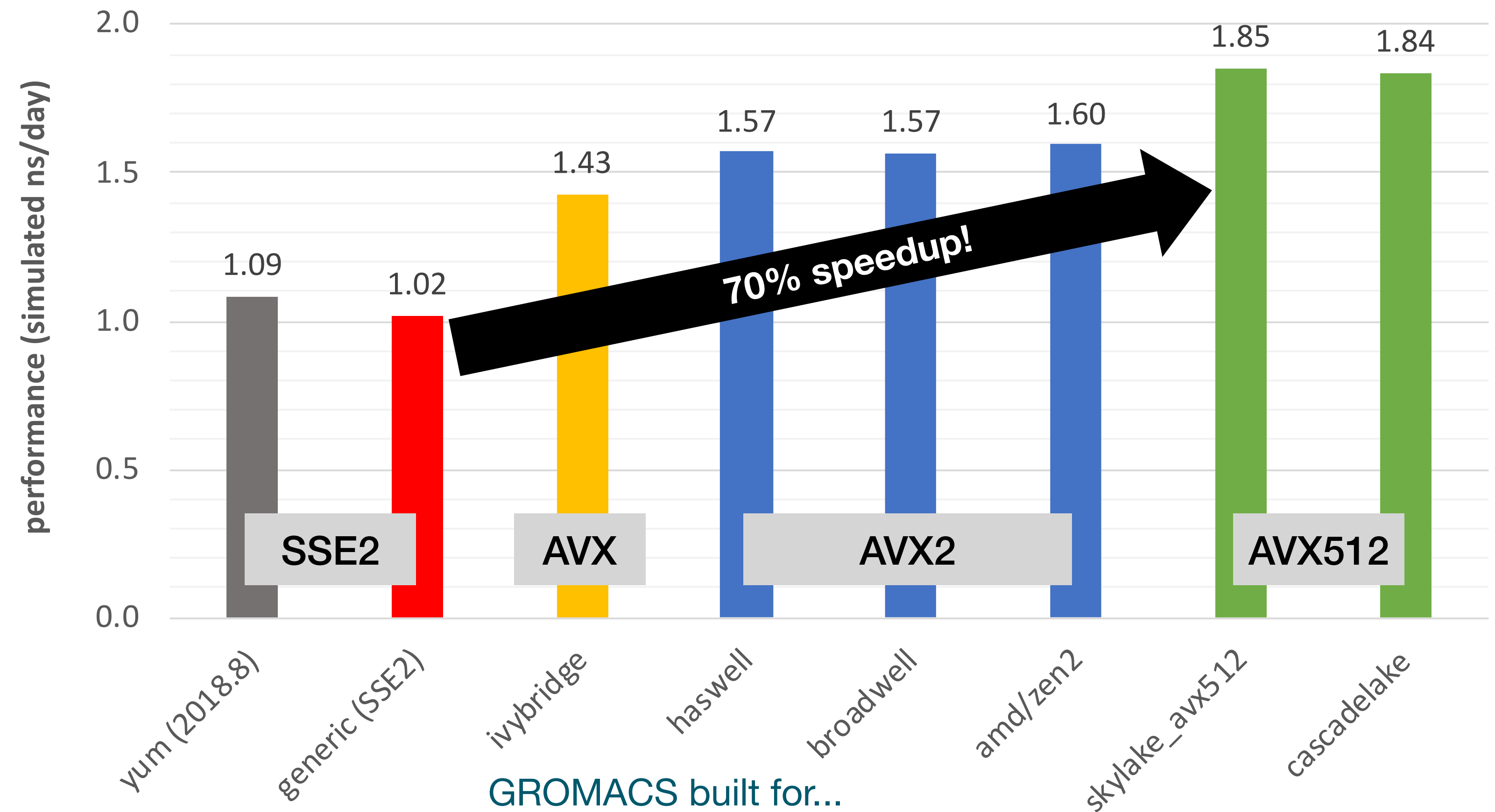
There's a storm coming... or are we in it already?

- Explosion of open source scientific software in recent years
- Increasing variety in CPUs: Intel, AMD, Arm, POWER, RISC-V
- Various types of accelerators: NVIDIA & AMD GPUs, Intel Xe, ...
- Rise of the cloud: Amazon EC2, Microsoft Azure, Google, Oracle, ...
- In stark contrast: available manpower in HPC support teams...

Keeping the P in HPC

- **Software should be optimised for the system it will run on**
- Impact on performance is often significant for scientific software

- Example: GROMACS
(PRACE benchmark, Test Case B)
- Metric: (simulated) ns/day,
higher is better
- Test system: dual-socket
Intel Xeon Gold 6420
(Cascade Lake, 2x18 cores)



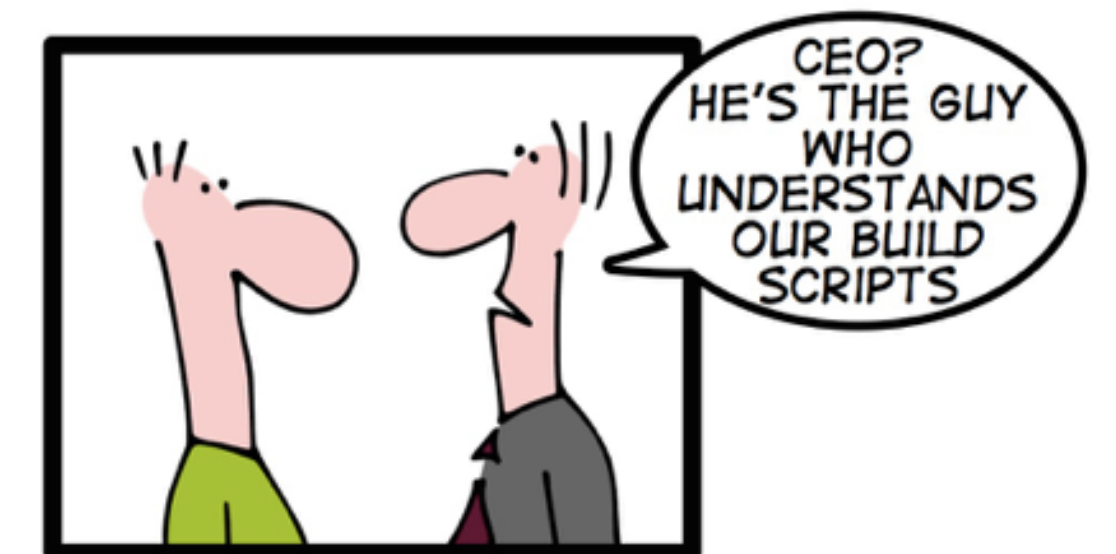
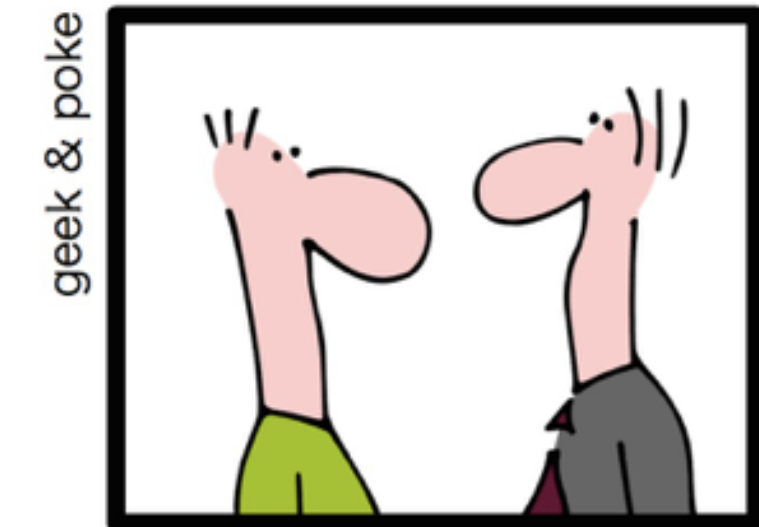
Getting Scientific Software Installed

```
INSTALL.SH  
#!/bin/bash  
  
pip install "$1" &  
easy_install "$1" &  
brew install "$1" &  
npm install "$1" &  
yum install "$1" & dnf install "$1" &  
docker run "$1" &  
pkg install "$1" &  
apt-get install "$1" &  
sudo apt-get install "$1" &  
steamcmd +app_update "$1" validate &  
git clone https://github.com/"$1"/"$1" &  
cd "$1";./configure;make;make install &  
curl "$1" | bash &
```

<https://xkcd.com/303>

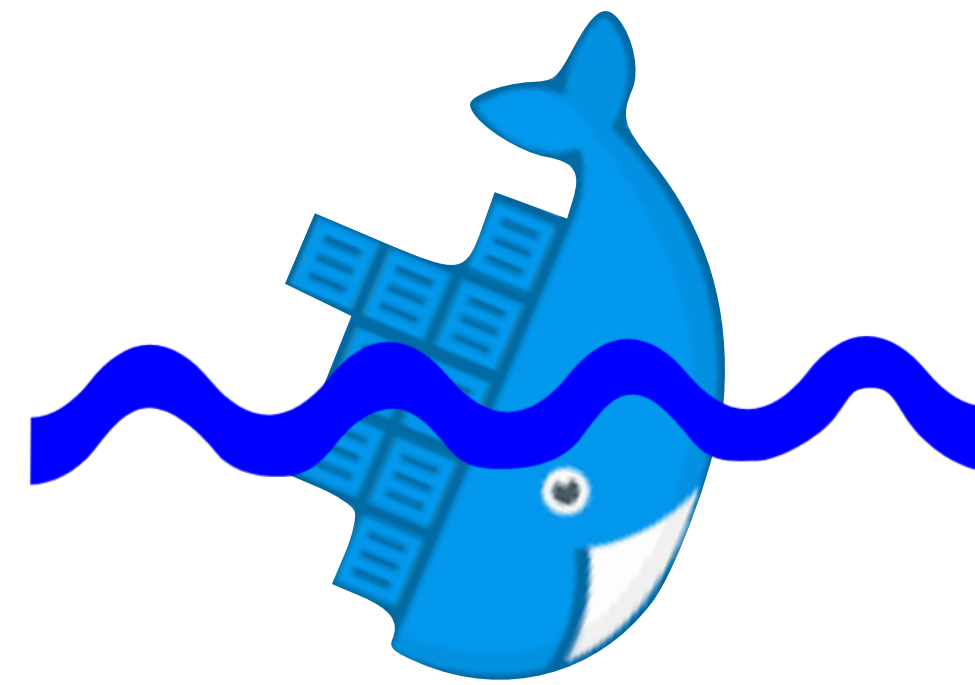


<https://xkcd.com/1654>



HOW TO BECOME INVALUABLE

What about... containers?



- Containers are (still) problematic in an HPC context
 - Integration with system resources & services is not a solved problem (MPI, GPUs, ...)
- "Native performance" really means "very little overhead"
 - That's very good, and a *requirement* for HPC, but not enough...
- Someone (you trust) has to build (and maintain) the container image you need!
 - Including the software you want to use, and for your type of system (x86_64, Arm64, ...)
 - Ideally, the image is properly optimised for the *specific* hardware you'll be using it on (performance vs "mobility of compute")
- **Containers are a symptom, not a cure...**



Scope & goals

- **Shared repository of scientific software installations**
- Collaborate, avoid duplicate work across HPC sites
- **Uniform way of providing software to researchers**
- Offer broad platform support (Linux, macOS, Windows)
- **Targets: laptops, personal workstations, HPC clusters, and the cloud**
- Support for different CPUs, interconnects, GPUs, etc.
- Focus on **performance**, automation, testing, tuning



Pick a live demo! (software)



1



GROMACS

version 2020.1

Molecular dynamics simulation

PRACE benchmark

2



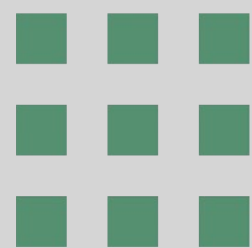
TensorFlow

version 2.3.1

Machine learning

Beginner tutorial (MNIST)

3



OpenFOAM

version 8 (openfoam.org)

Computational Fluid Dynamics (CFD) simulation

motorBike tutorial case

4



Bioconductor

version 3.11 (on top of R 4.0)

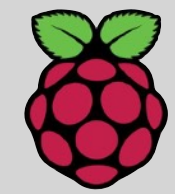
bioinformatics

Playing with DNA data

Pick a live demo! (hardware)



A



Raspberry Pi cluster

4x model 3B+



quad-core **Arm** Cortex-A53, 1GB RAM

64-bit Raspberry Pi OS

native CernVM-FS (built from source)



Amazon EC2

c6g.2xlarge instance type

8-core **Arm** Graviton 2, 8GB RAM

RHEL 8.2

CernVM-FS via **Singularity** container

B



Microsoft Azure

Standard_D8s_v3 instance

8-core Intel Xeon (**different CPU generations**)

Ubuntu 18.04

native CernVM-FS (installed via apt)

C



HPC-UGent cluster

AMD Rome CPUs

2x 48-core EPYC 7552, 250GB RAM

Red Hat Enterprise Linux 8.2

CernVM-FS via **Singularity** container

D



Spoiler demo



GROMACS



E E S S I

EUROPEAN ENVIRONMENT FOR
SCIENTIFIC SOFTWARE INSTALLATIONS

EESSI is powered by FOSS



CernVM-FS

- Software distribution service
- Scalable, read-only
- Mount filesystem over HTTP

<https://cernvm.cern.ch/fs>



gentoo

- Linux distribution, installs from source
- Prefix project: install packages in <prefix>
- Supports x86_64, Arm, ... & Linux, macOS

<https://wiki.gentoo.org/wiki/Project:Prefix>



easybuild

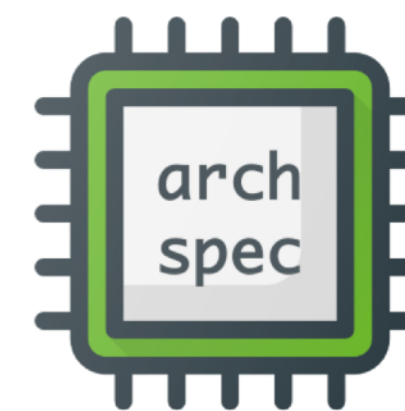
- Installation tool for scientific software
- Optimises for build host (by default)
- Supports over 2,000 software pkgs

<https://easybuilders.github.io/easybuild>

Lmod

- Environment modules tool (written in Lua)
- Intuitive access to software installations
- Multiple software versions side-by-side

<https://lmod.readthedocs.io>



- Python library
- Detect CPU family
- Compare CPUs (compatibility)

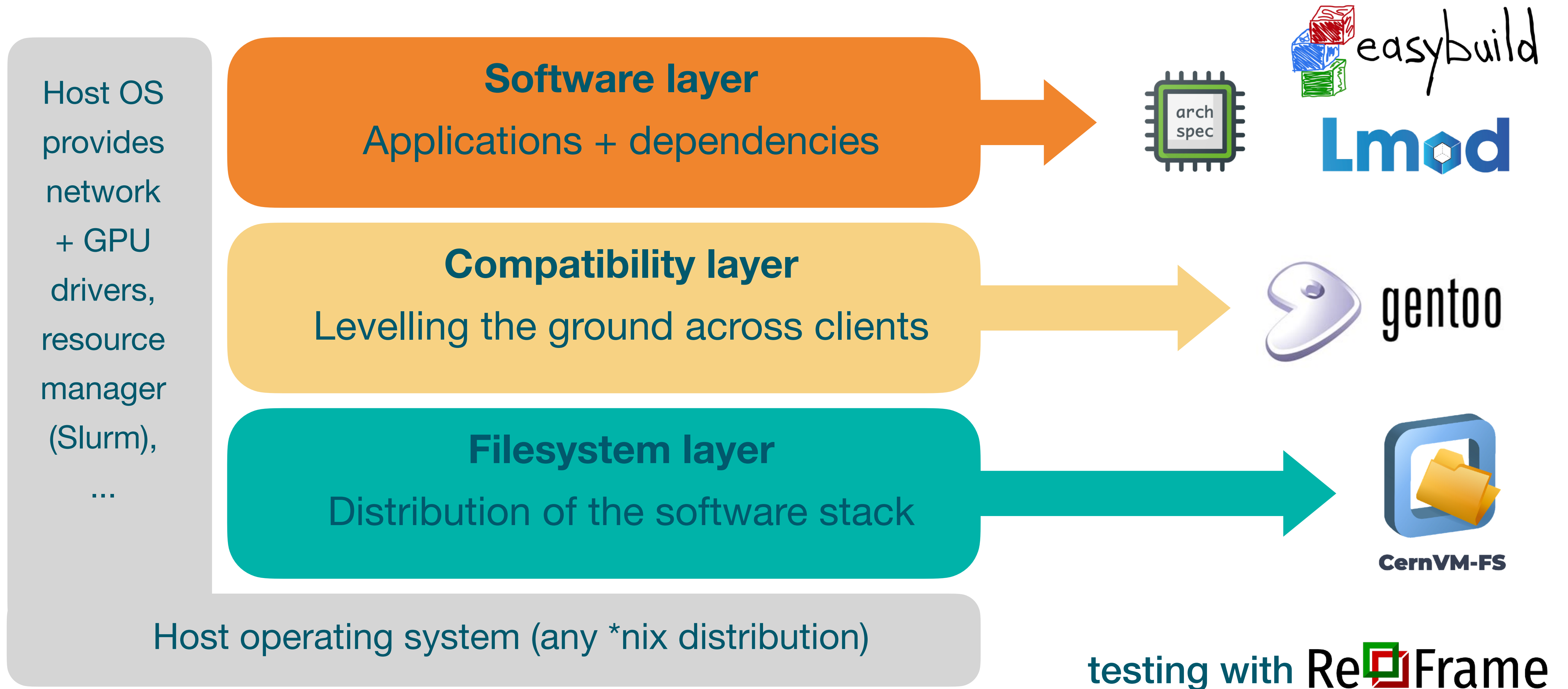
<https://github.com/archspec>

ReFrame

- regression testing framework for HPC
- tests are implemented in Python

<https://reframe-hpc.readthedocs.io>

High-level overview of EESSI project

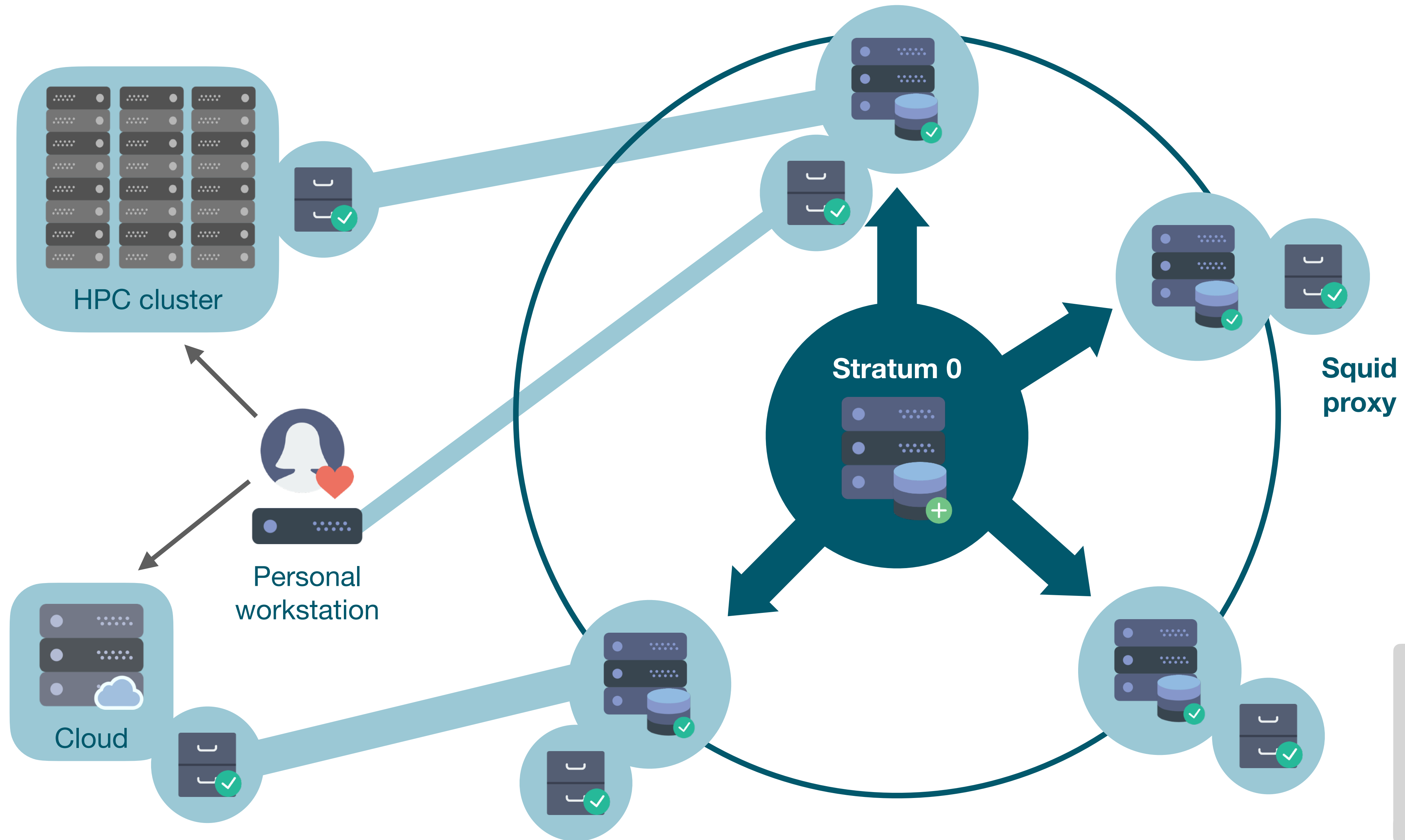


Inspiration for EESSI



- EESSI concept is **heavily** inspired by Compute Canada software stack
- Shared across 5 major national systems in Canada + a bunch of smaller ones
- 3 layers: CernVM-FS / ~~Nix~~ Gentoo Prefix / EasyBuild + Lmod
- See paper by Maxime Boissonneault & co at PEARC'19 (PDF available [here](#))
“Providing a Unified Software Environment for Canada’s National Advanced Computing Centers”
- See also Maxime’s talk at 5th EasyBuild User Meeting ([slides](#) - [recorded talk](#))
and the Compute Canada [documentation](#)

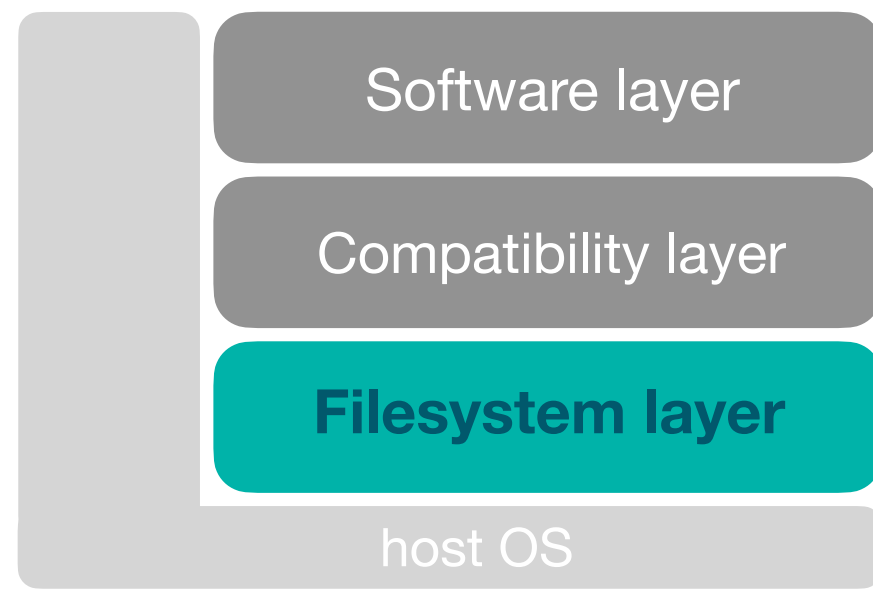
EESSI "network"



powered by



CernVM-FS



EESSI "network"



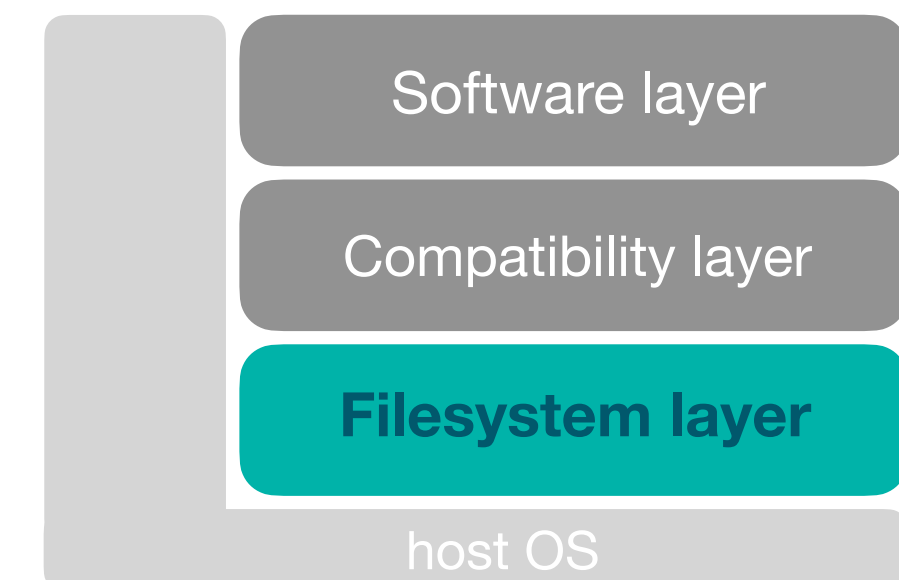
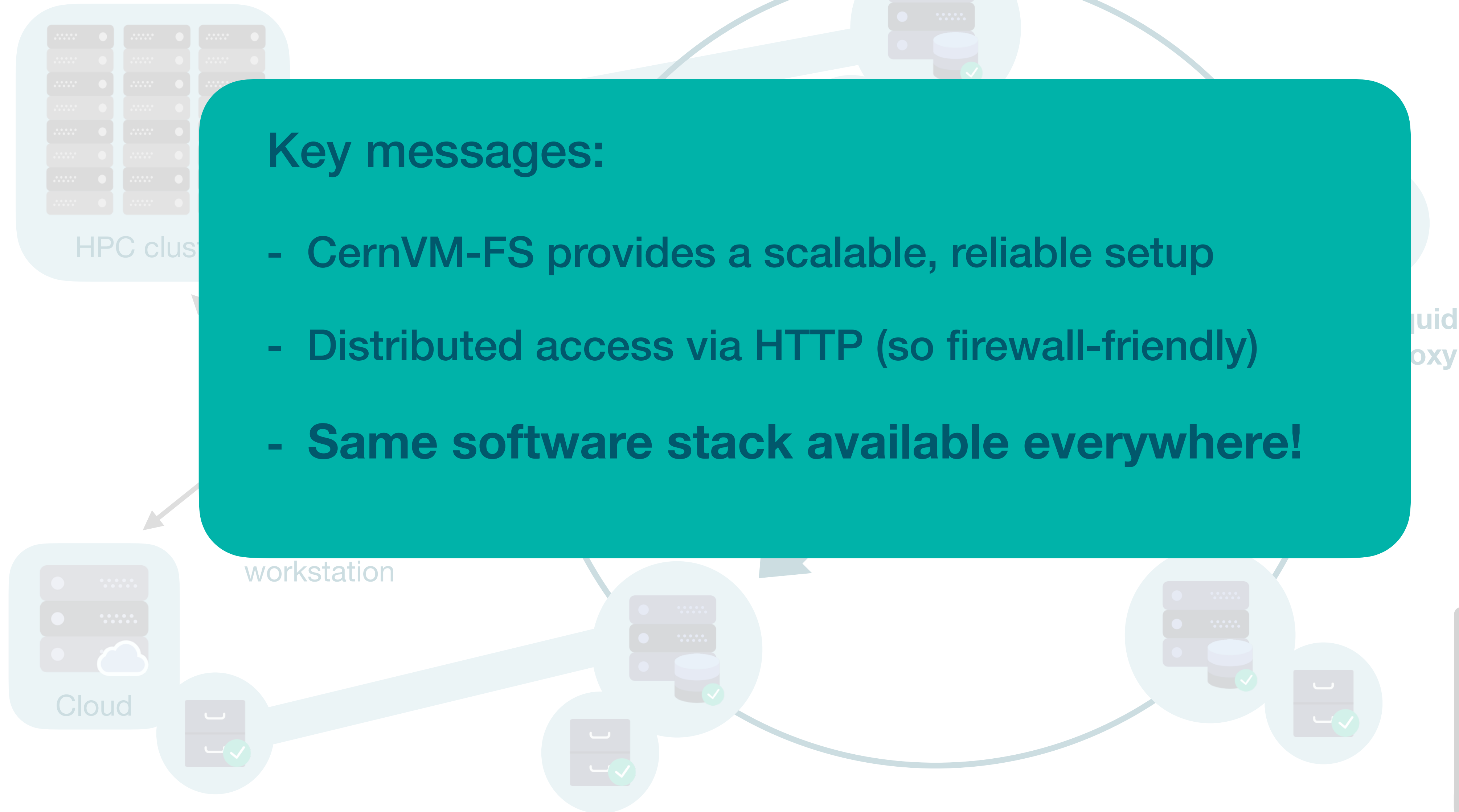
Key messages:

- CernVM-FS provides a scalable, reliable setup
- Distributed access via HTTP (so firewall-friendly)
- **Same software stack available everywhere!**

powered by



CernVM-FS

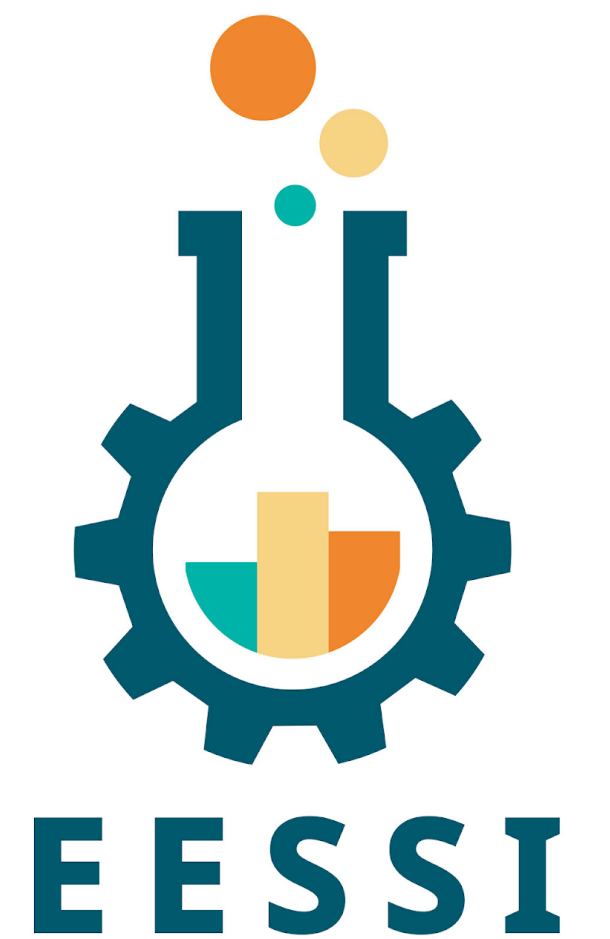


Compatibility layer

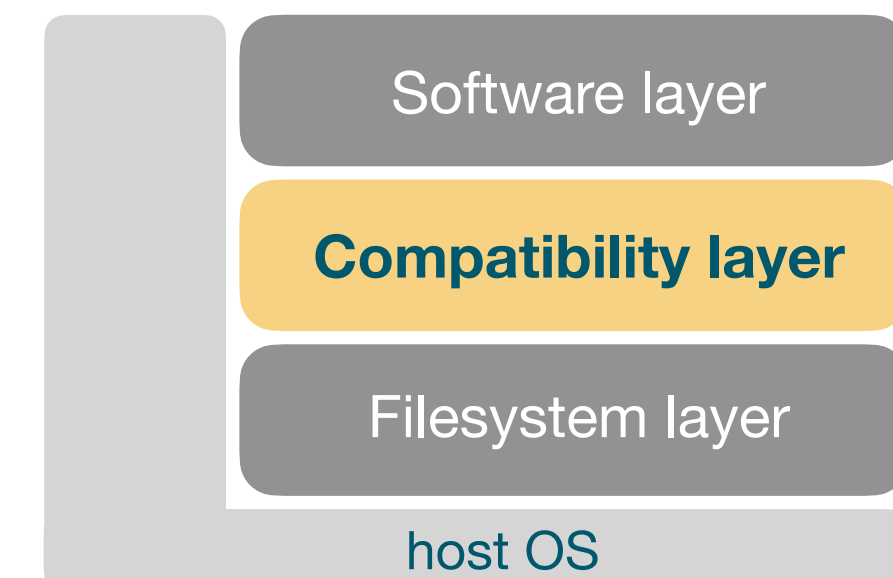
- Set of tools & libraries installed in non-standard location
- Using Gentoo's package manager Portage
- Limited to low-level stuff, incl. glibc
- Only targets a supported processor **family** (x86_64, Arm64)
- **Levels the ground for different client operating systems** (Linux distros, macOS)
- Currently in pilot repository:

```
/cvmfs/pilot.eessi-hpc/2020.10/compat/aarch64
```

```
/cvmfs/pilot.eessi-hpc/2020.10/compat/x86_64
```

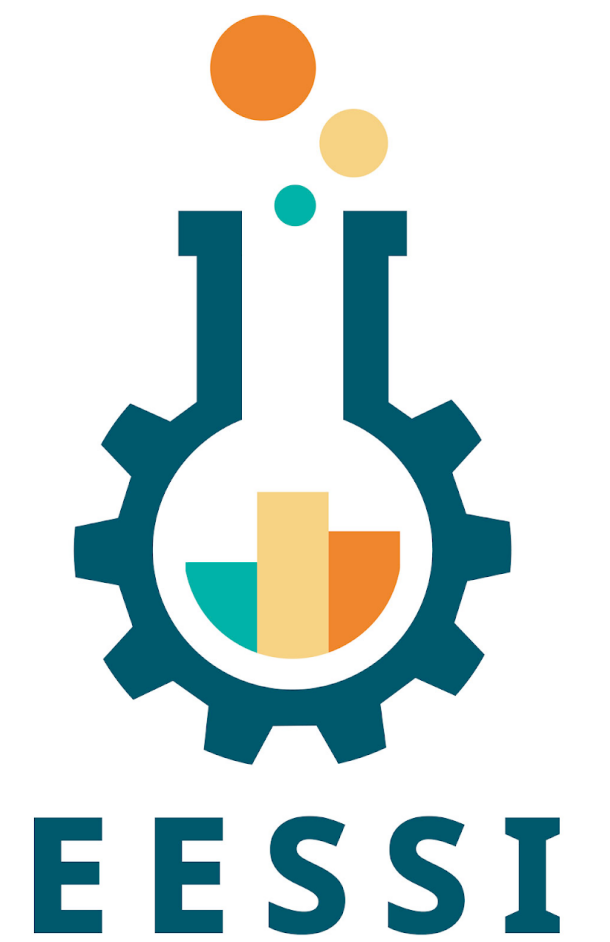


powered by

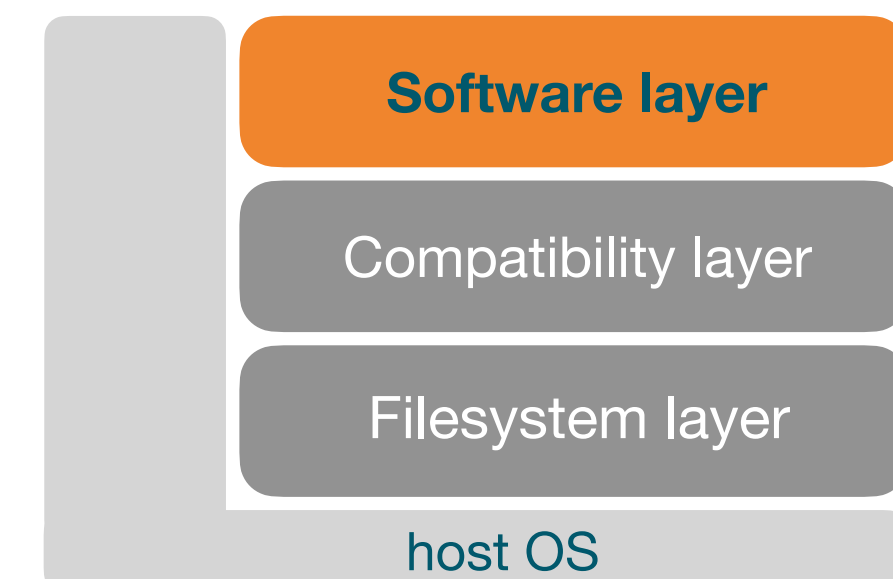
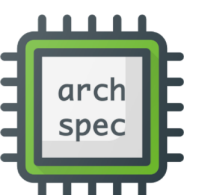
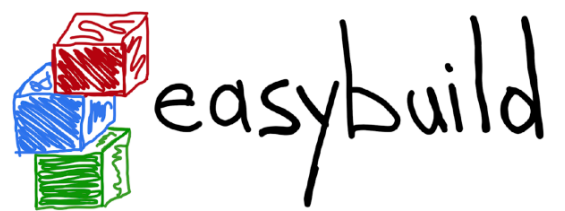


Software layer

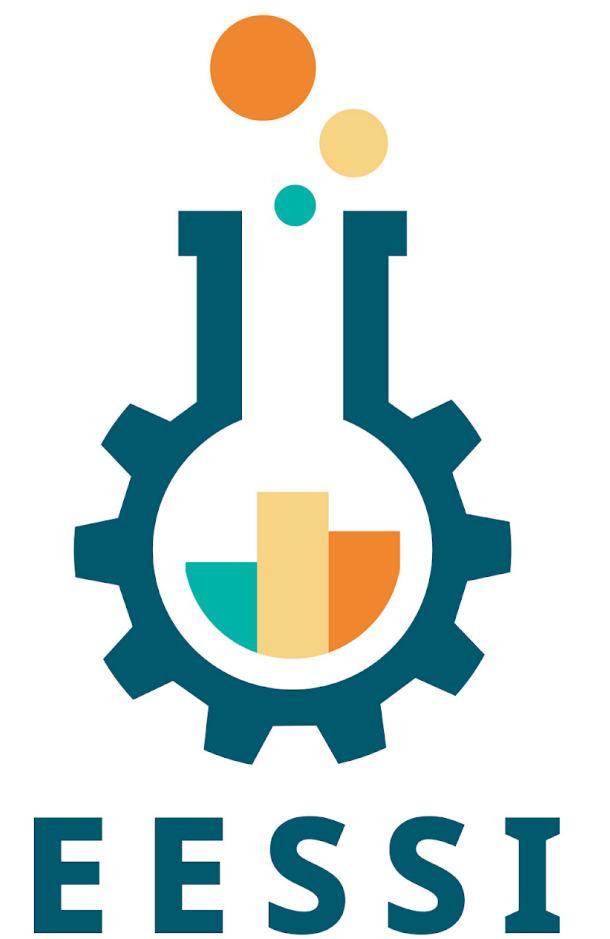
- Provides scientific software applications, libraries, and dependencies
- **Optimised for specific CPU microarchitectures** (Intel Haswell, ...)
- **Leverages libraries from compatibility layer** (not from host OS)
- Installed with EasyBuild, incl. environment module files
- Lmod environment modules tool is used to access installations
- Different subdirectories: one per target CPU microarchitecture
- **Best subdirectory for host is picked automatically** via archspec



powered by



Current status: pilot repository

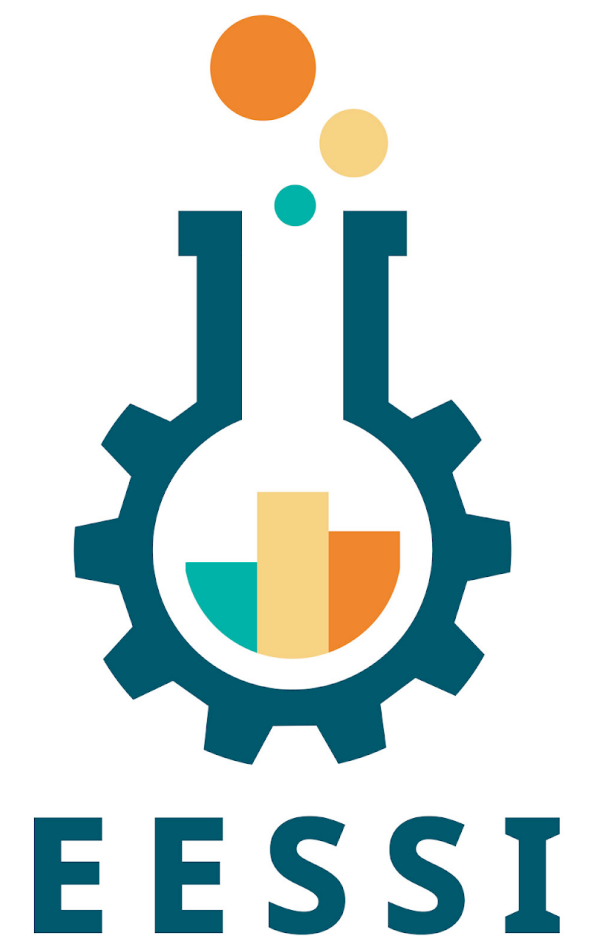


- Ansible playbooks, scripts, docs at <https://github.com/eessi>
- Stratum 0 @ Univ. of Groningen + two Stratum 1 servers
- Compatibility layer for both x86_64 and aarch64 (only Linux clients, for now)
- Software (CPU-only): Bioconductor, GROMACS, OpenFOAM, TensorFlow
- Hardware targets:
 - x86_64/generic, intel/haswell, intel/skylake_avx512, amd/zen2
 - aarch64/generic, aarch64/graviton2

**NOT FOR
PRODUCTION USE!**

Try it yourself: <https://eessi.github.io/docs/pilot>

From zero to science in 3 steps



Step 1: Access the (pilot) EESSI CernVM-FS repository

Option 1: Native CernVM-FS installation (requires admin privileges)

- Install CernVM-FS, see <https://cernvm.cern.ch/fs>
- Also install configuration files for EESSI repositories, available at <https://github.com/EESSI/filesystem-layer/releases>
- Run "cvmfs_config setup"

Option 2: Use a Singularity container (no admin rights needed)

- CernVM-FS repositories can be mounted via Singularity's `--fusemount` option
- See detailed walkthrough at <https://eessi.github.io/docs/pilot>

From zero to science in 3 steps

Step 1: Access the (pilot) EESSI CernVM-FS repository

Option 1 (example): native CernVM-FS installation on fresh (x86_64) RHEL 8.2 system



```
# install CernVM-FS (see https://cernvm.cern.ch/fs/)
sudo yum install -y https://ecsft.cern.ch/dist/cvmfs/cvmfs-release/cvmfs-release-latest.noarch.rpm
sudo yum install -y cvmfs

# install CernVM-FS configuration files for EESSI repositories (see https://github.com/EESSI/filesystem-layer)
wget https://github.com/EESSI/filesystem-layer/releases/download/v0.2.3/cvmfs-config-eessi-0.2.3-1.noarch.rpm
sudo yum install -y cvmfs-config-eessi-0.2.3-1.noarch.rpm

# create local CernVM-FS configuration file (direct access, no proxy; 10GB for CernVM-FS cache)
sudo bash -c "echo 'CVMFS_HTTP_PROXY=DIRECT' > /etc/cvmfs/default.local"
sudo bash -c "echo 'CVMFS_QUOTA_LIMIT=10000' >> /etc/cvmfs/default.local"

# set up CernVM-FS
sudo cvmfs_config setup

# access EESSI pilot repository
ls /cvmfs/pilot.eessi-hpc.org/2020.10
```

From zero to science in 3 steps

Step 1: Access the (pilot) EESSI CernVM-FS repository

Option 2 (example): use Singularity to run Docker container to access EESSI



```
# configure Singularity (bind mounts + home directory)
mkdir -p /tmp/$USER/{var-lib-cvmfs,var-run-cvmfs,home}
export SINGULARITY_BIND="/tmp/$USER/var-run-cvmfs:/var/run/cvmfs,/tmp/$USER/var-lib-cvmfs:/var/lib/cvmfs"
export SINGULARITY_HOME="/tmp/$USER/home:/home/$USER"

# values to pass to --fusemount (EESSI config + pilot repositories)
export EESSI_CONFIG="container:cvmfs2 cvmfs-config.eessi-hpc.org /cvmfs/cvmfs-config.eessi-hpc.org"
export EESSI_PILOT="container:cvmfs2 pilot.eessi-hpc.org /cvmfs/pilot.eessi-hpc.org"

# minimal Docker container from Docker Hub (includes CernVM-FS + EESSI configuration files)
export DOCKER_IMAGE="docker://eessi/client-pilot:centos7-$(uname -m)-2020.10"

# start shell in Singularity container (ignore the scary looking 'setxattr' warnings, they're harmless)
singularity shell --fusemount "$EESSI_CONFIG" --fusemount "$EESSI_PILOT" $DOCKER_IMAGE

# access EESSI pilot repository
ls /cvmfs/pilot.eessi-hpc.org/2020.10
```

From zero to science in 3 steps



Step 2: Set up your environment by sourcing the EESSI `init` script

```
# source the EESSI init script to set up your environment
$ source /cvmfs/pilot.eessi-hpc.org/2020.10/init/bash
Found EESSI pilot repo @ /cvmfs/pilot.eessi-hpc.org/2020.10!
Using x86_64/intel/haswell as software subdirectory.
Initializing Lmod...
Prepending /cvmfs/pilot.eessi-hpc.org/2020.10/software/x86_64/intel/haswell/modules/all to $MODULEPATH...

Environment set up to use EESSI pilot software stack, have fun!

[EESSI pilot 2020.10] $ echo $EESSI_PREFIX
/cvmfs/pilot.eessi-hpc.org/2020.10

[EESSI pilot 2020.10] $ echo $EESSI_SOFTWARE_SUBDIR
x86_64/intel/haswell
```

From zero to science in 3 steps

Step 3: Load the modules for the software you want to use, and go!



```
# check which modules are available
[EESSI pilot 2020.10] $ module avail gromacs

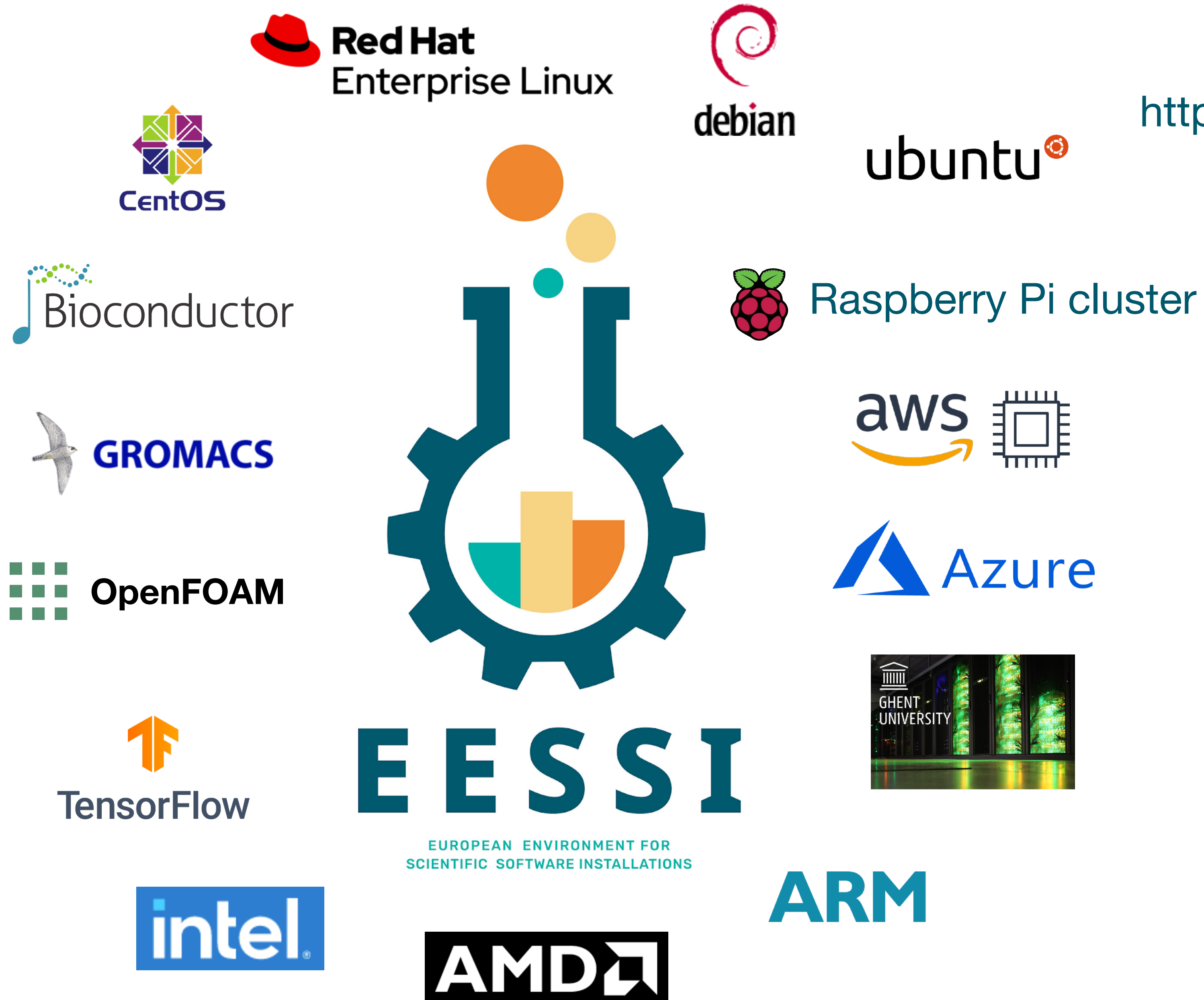
----- /cvmfs/pilot.eessi-hpc.org/2020.10/software/x86_64/intel/haswell/modules/all -----
GROMACS/2020.1-foss-2020a-Python-3.8.2

# load the module(s) for the software you want to use
[EESSI pilot 2020.10] $ module load GROMACS

# ready to compute!
[EESSI pilot 2020.10] $ gmx mdrun -s ion_channel.tpr -maxh 0.50 -rethway -noconfout -nstps 1000
```

Demo time!

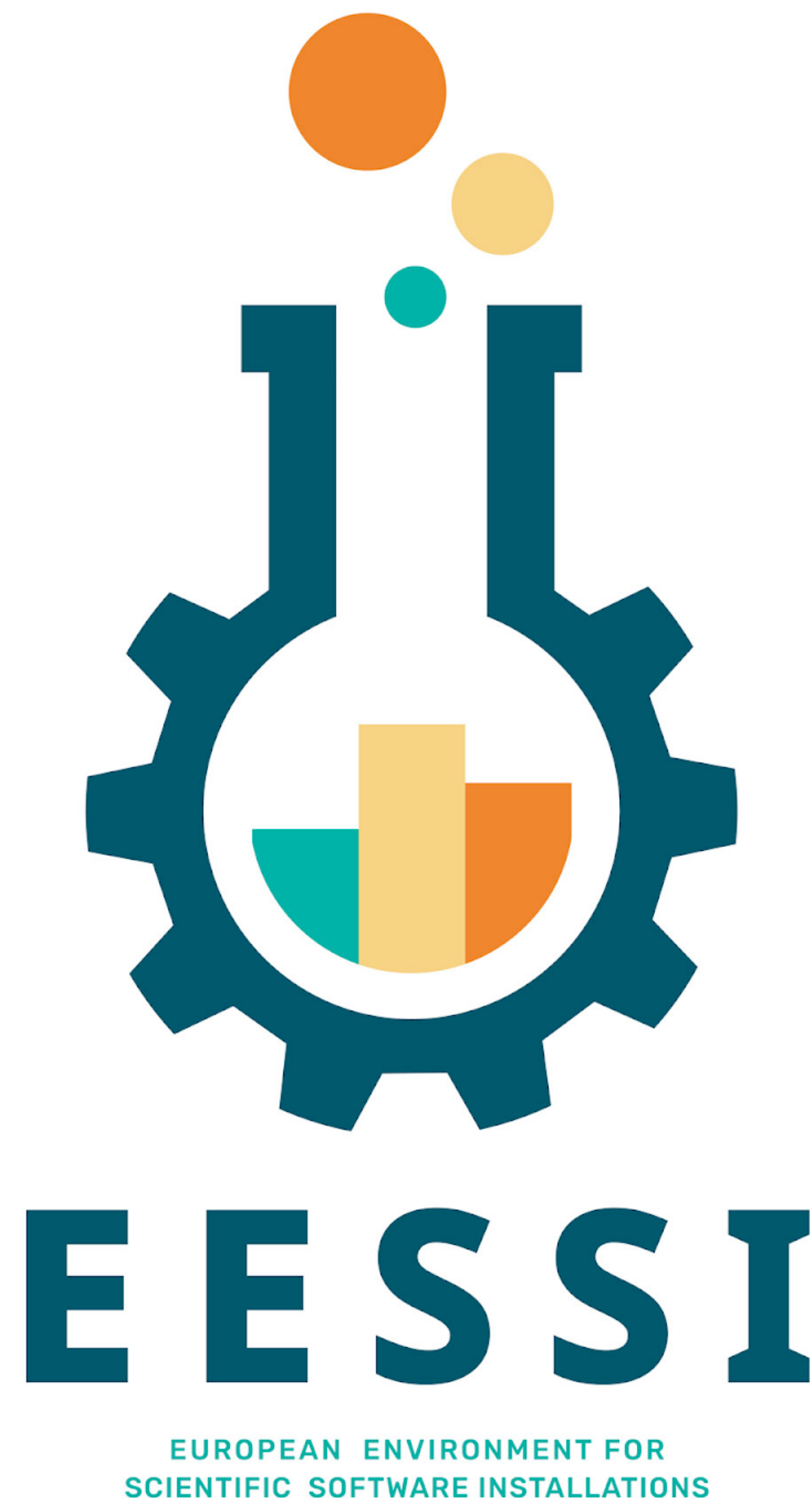
<https://github.com/boegel/eessi-demo>



Future work



- Further improve pilot EESSI repository (monthly revisions)
- Identify problems, and fix them
- **Automate** deployment of different EESSI layers (Ansible, Terraform, ...)
- **Testing** (with ReFrame) + continuous integration in GitHub Actions
- Let developers of scientific software validate the installation of *their* software
- Also support macOS, POWER, GPUs + add more software
- Solicit more manpower, get project funded to make it sustainable
- Work towards production setup...



Kenneth Hoste

kenneth.hoste@ugent.be

[@kehoste](#) (Twitter)

Website: <https://www.eessi-hpc.org>

Join our mailing list & Slack channel

<https://www.eessi-hpc.org/join>

Documentation: <https://eessi.github.io/docs>

GitHub: <https://github.com/eessi>

Twitter: [@eessi_hpc](#)

Monthly online meetings (first Thursday, 2pm CET)



International Series of Online Research Software Events

<https://sorse.github.io>

(BACKUP SLIDES)

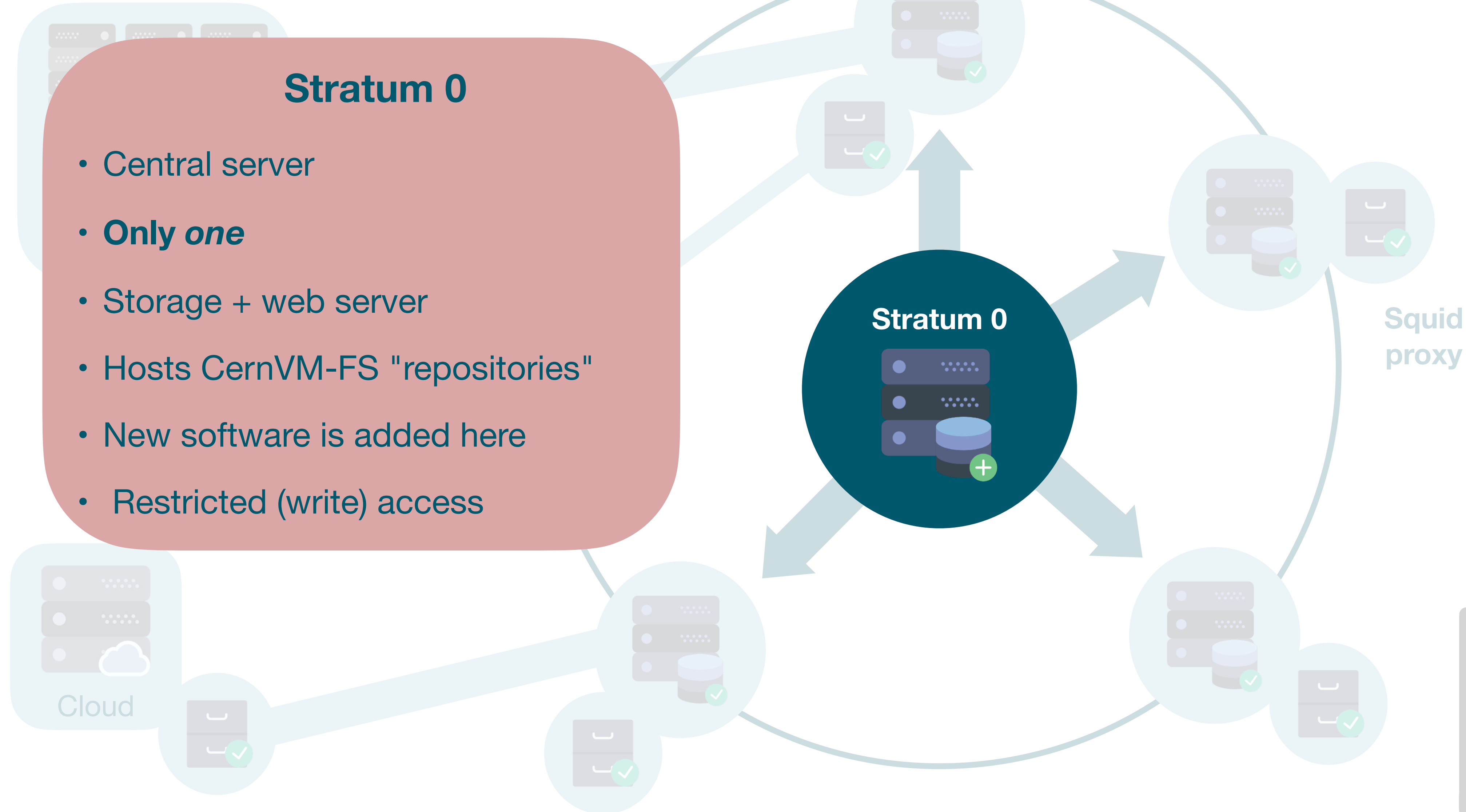
EESSI "network"



powered by



CernVM-FS



EESSI "network"

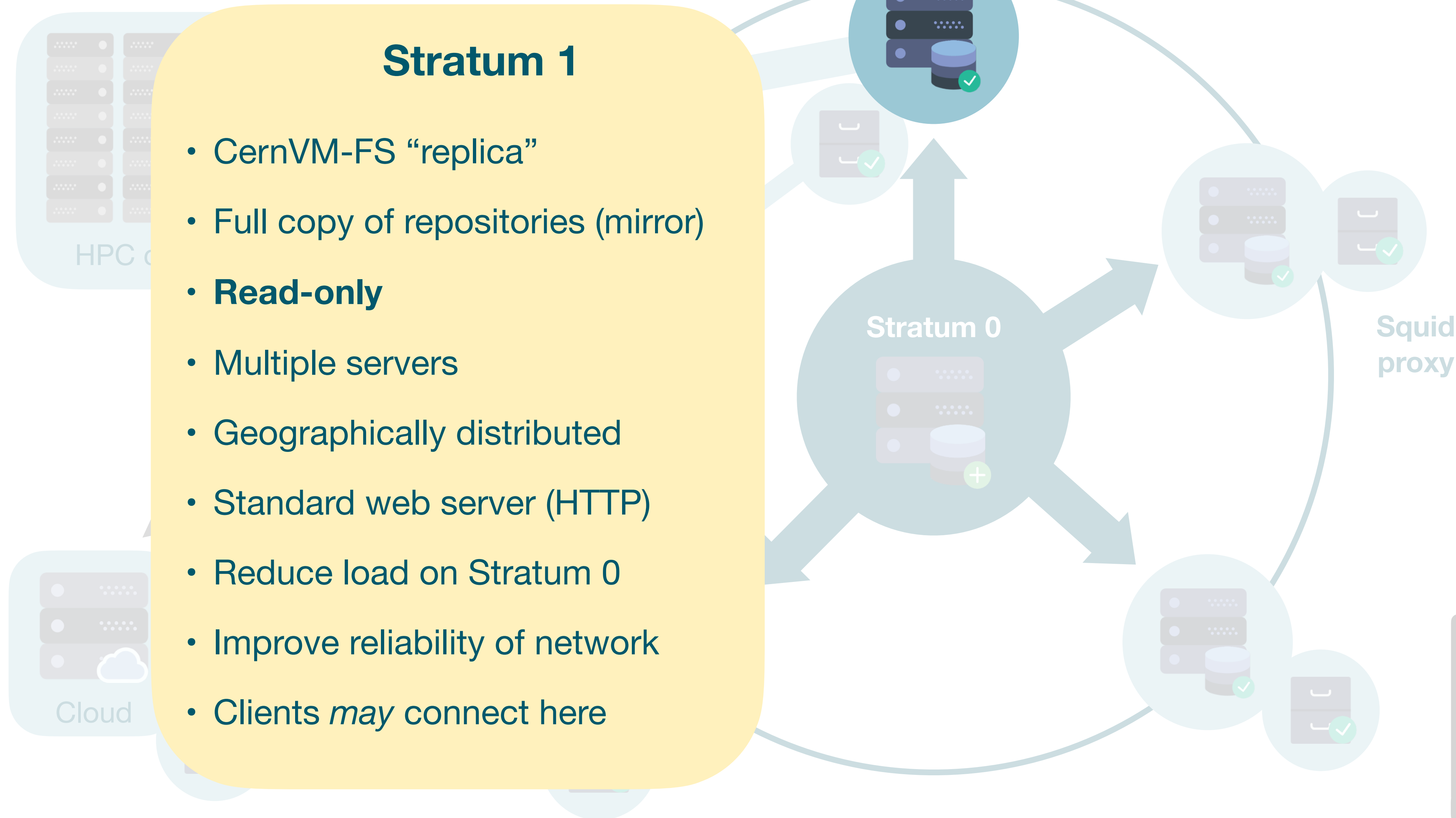


powered by

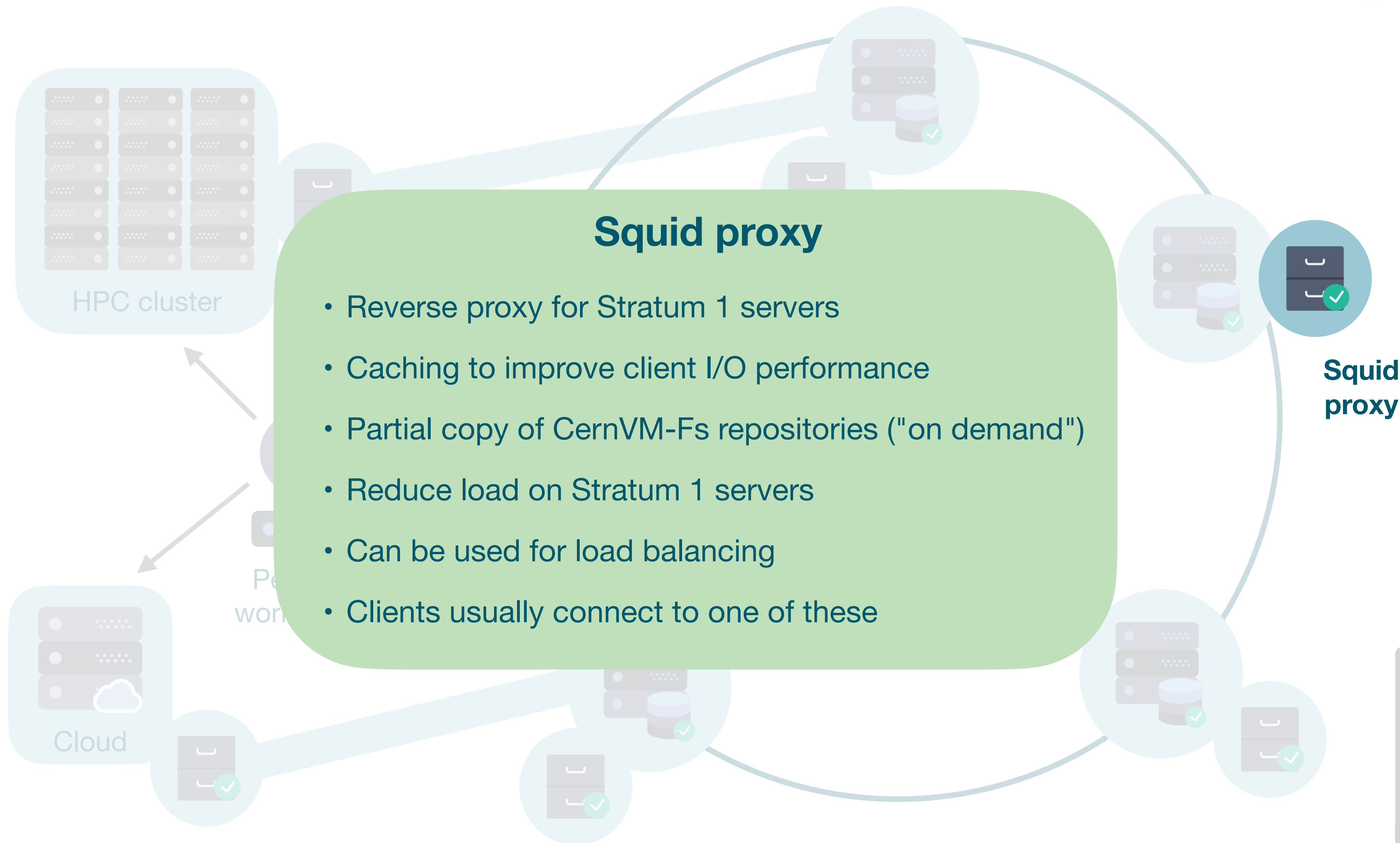


CernVM-FS

- ## Stratum 1
- CernVM-FS “replica”
 - Full copy of repositories (mirror)
 - **Read-only**
 - Multiple servers
 - Geographically distributed
 - Standard web server (HTTP)
 - Reduce load on Stratum 0
 - Improve reliability of network
 - Clients *may* connect here



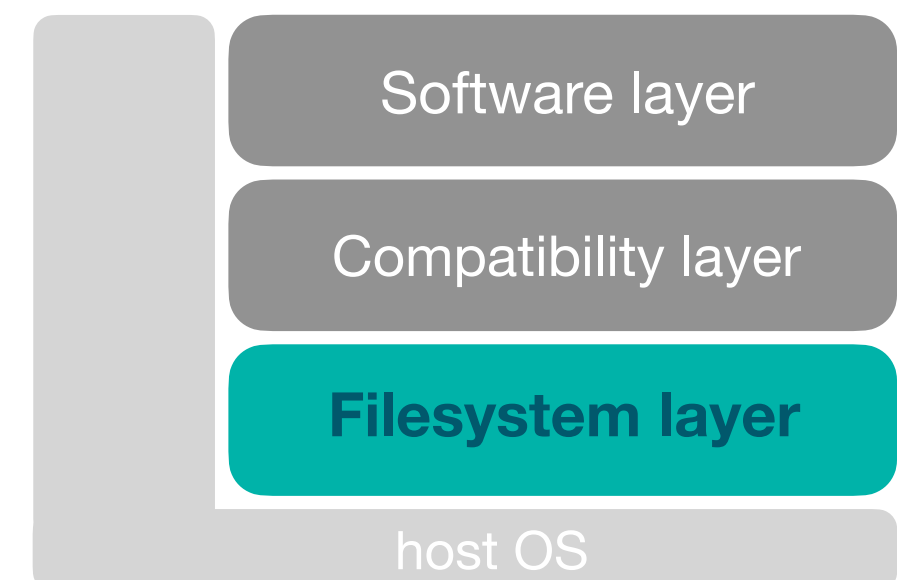
EESSI "network"



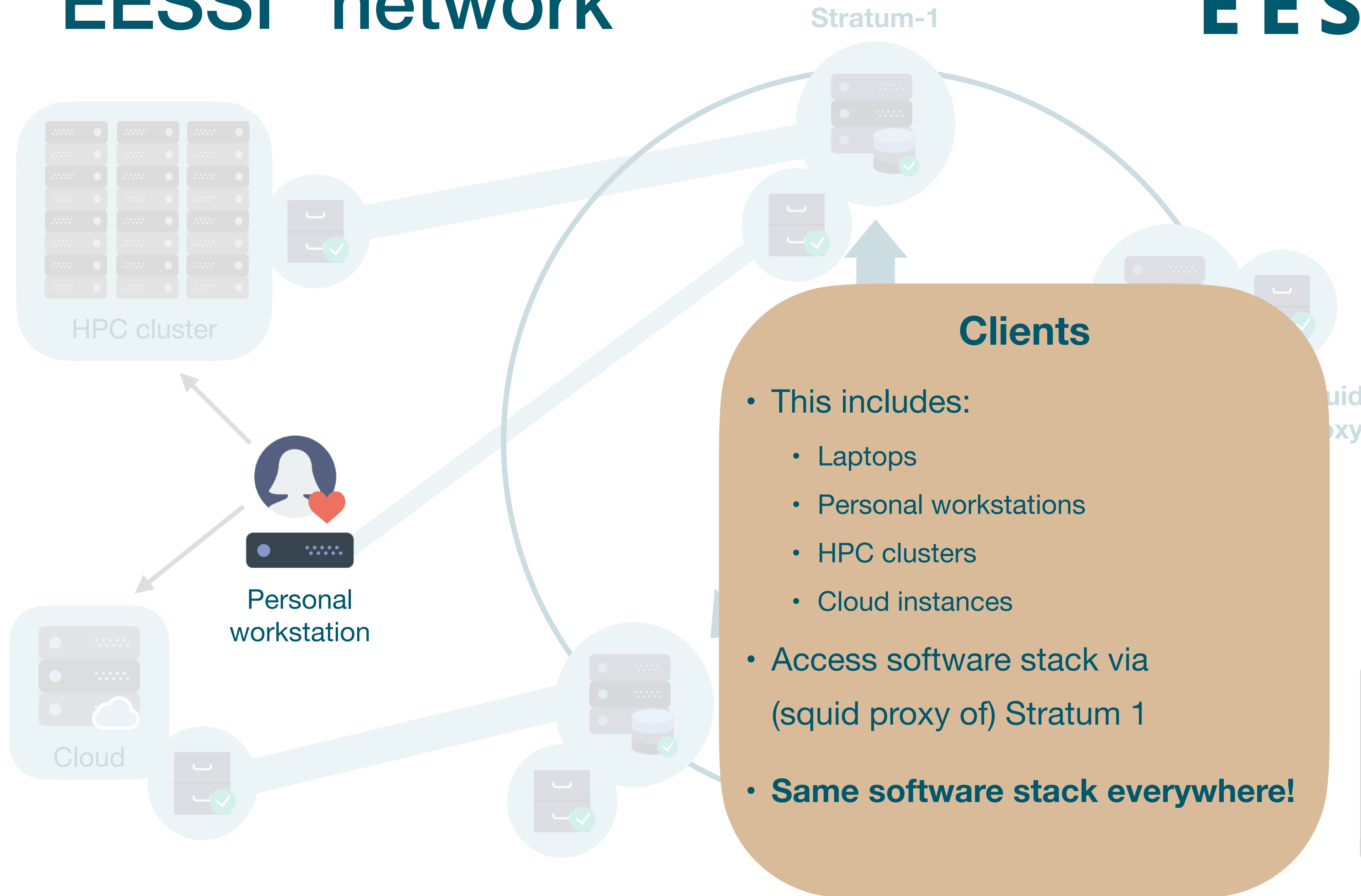
powered by



CernVM-FS



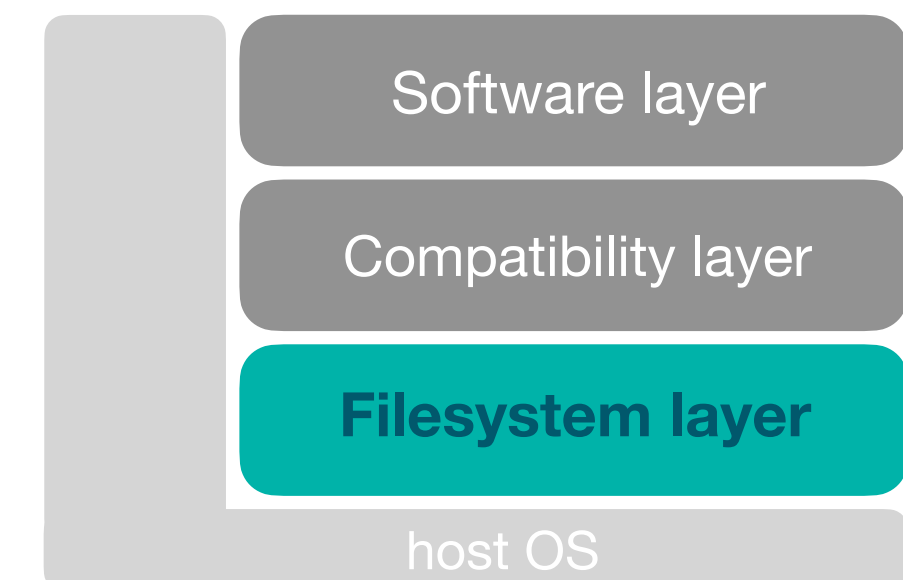
EESSI "network"



powered by

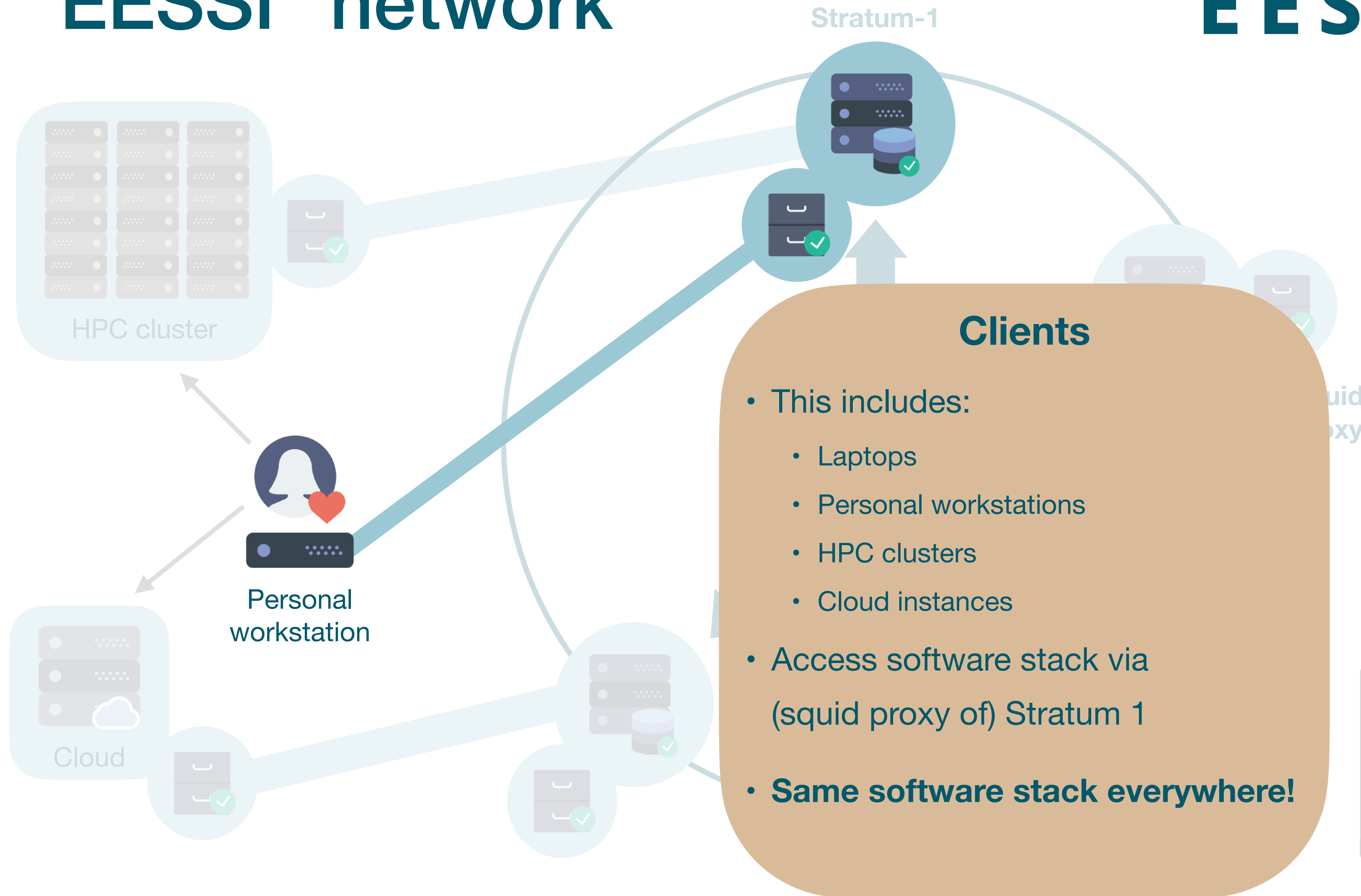


CernVM-FS



- This includes:
 - Laptops
 - Personal workstations
 - HPC clusters
 - Cloud instances
- Access software stack via (squid proxy of) Stratum 1
- **Same software stack everywhere!**

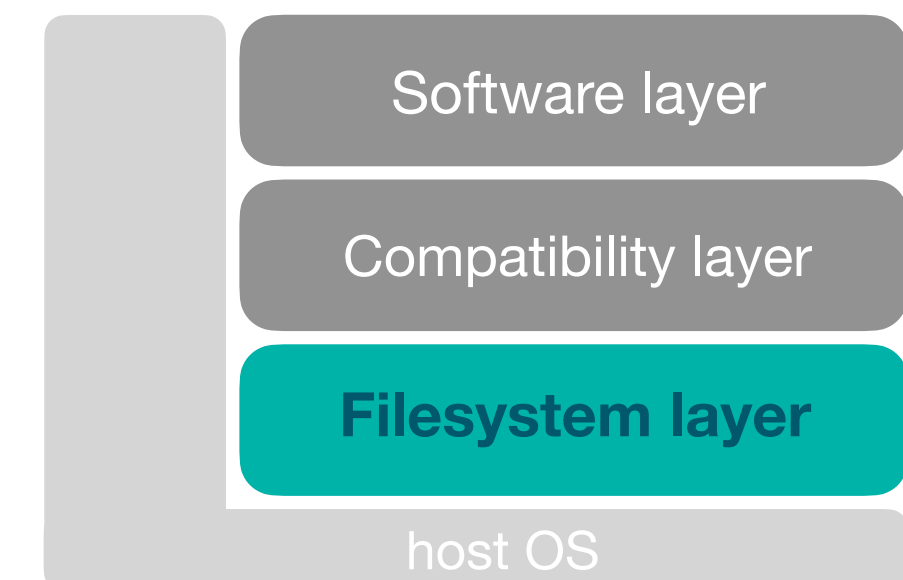
EESSI "network"



powered by

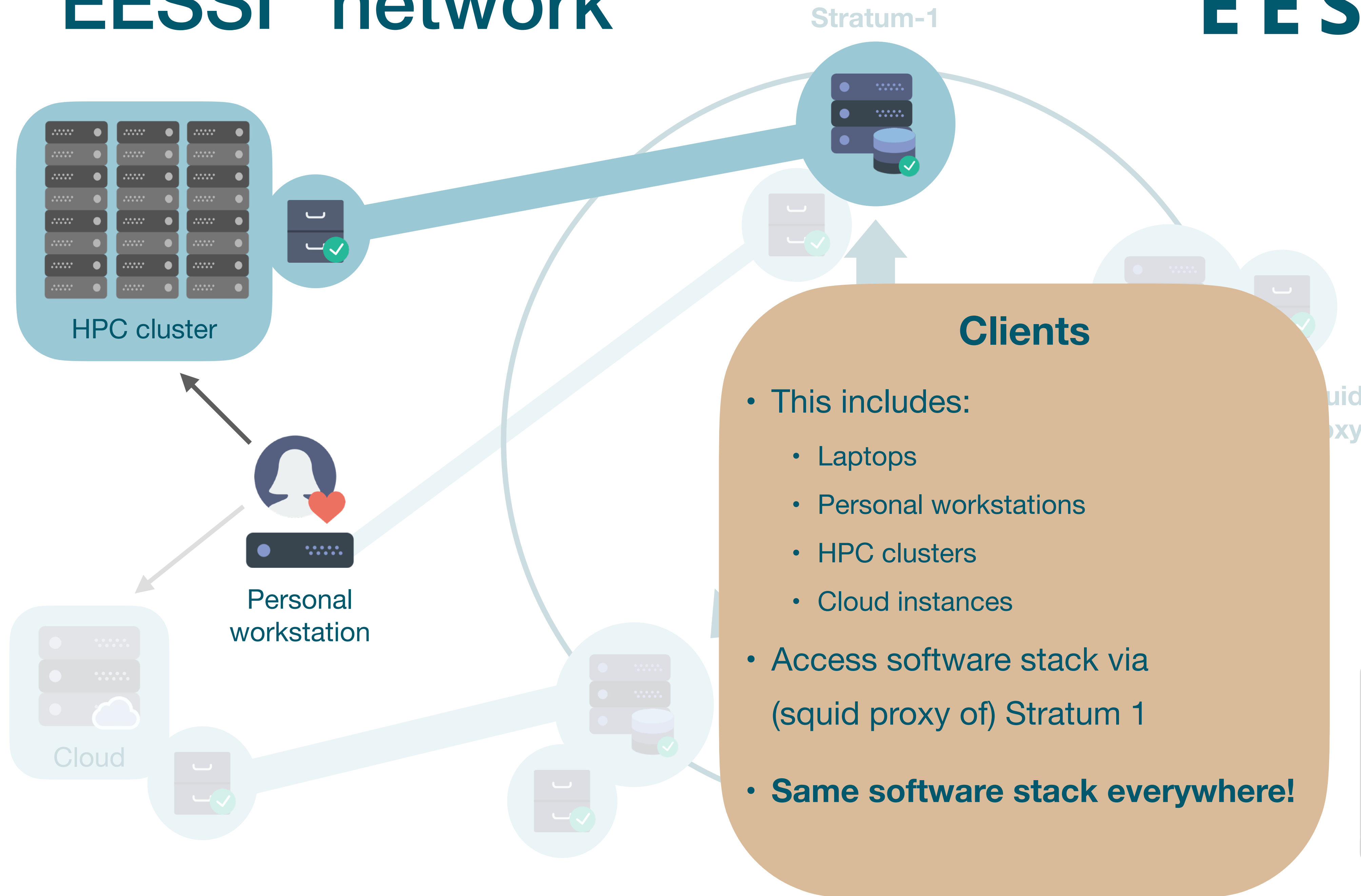


CernVM-FS



- This includes:
 - Laptops
 - Personal workstations
 - HPC clusters
 - Cloud instances
- Access software stack via (squid proxy of) Stratum 1
- **Same software stack everywhere!**

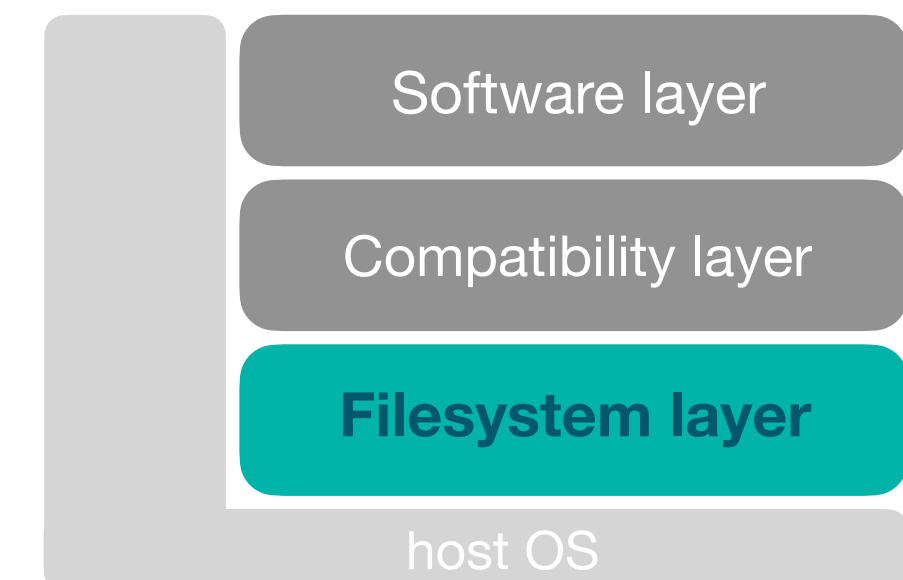
EESSI "network"



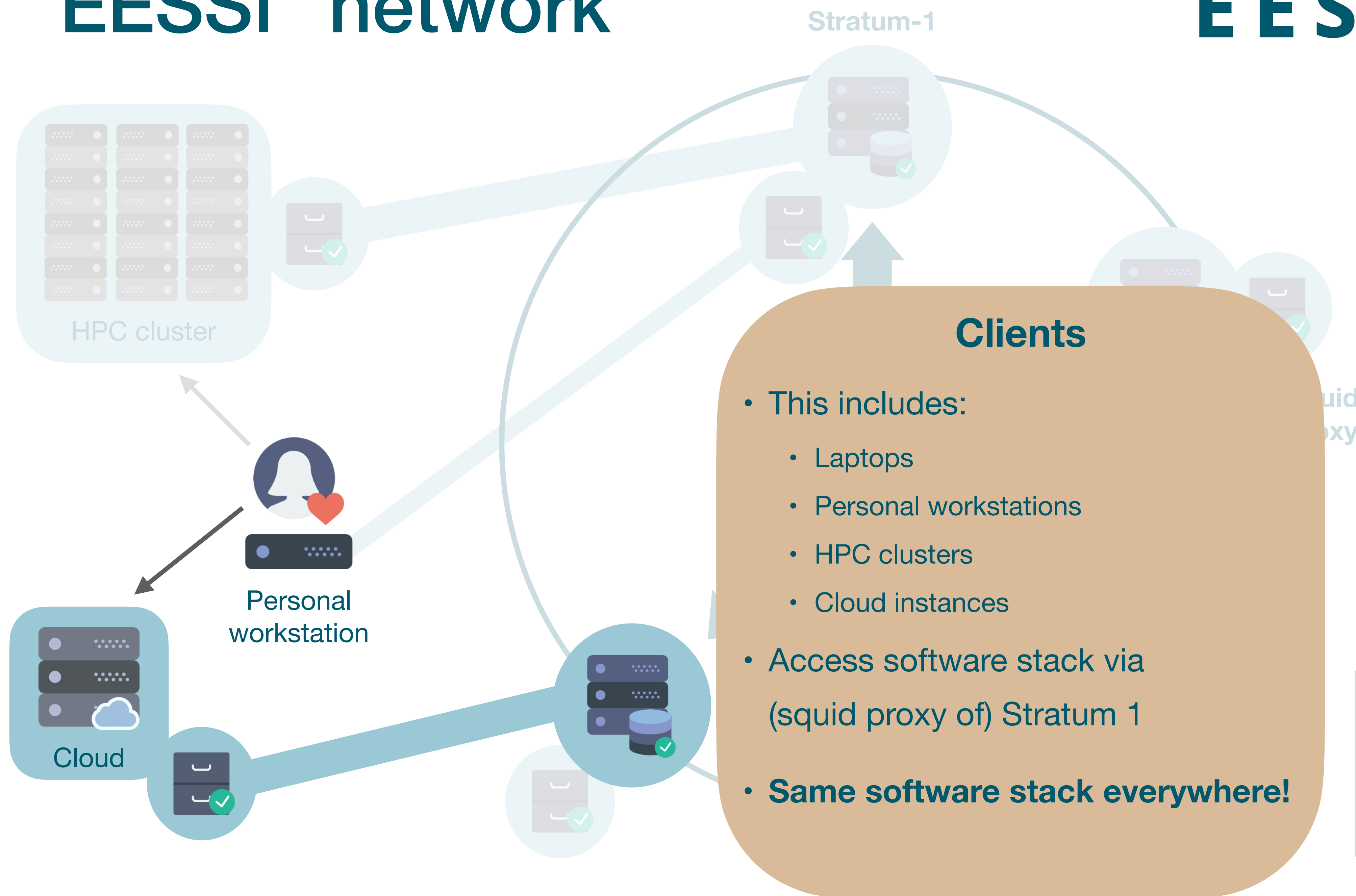
powered by



CernVM-FS



EESSI "network"



- This includes:
 - Laptops
 - Personal workstations
 - HPC clusters
 - Cloud instances
- Access software stack via (squid proxy of) Stratum 1
- **Same software stack everywhere!**

powered by

